炫舞2手机端\_小灵通工程文档

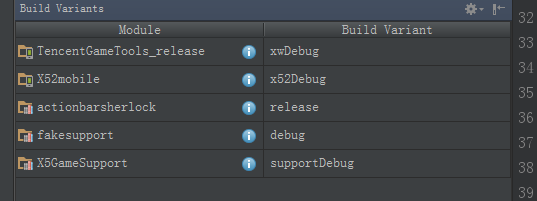
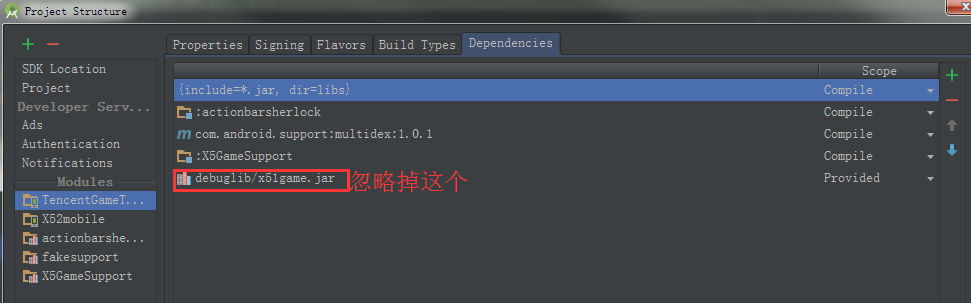
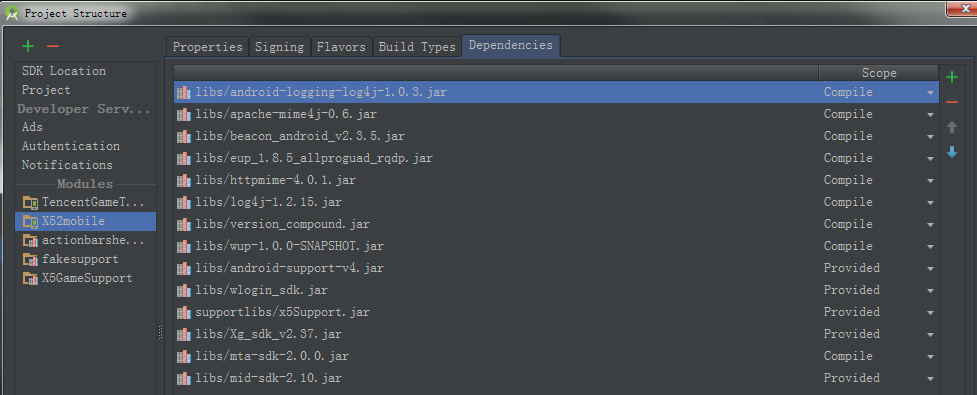
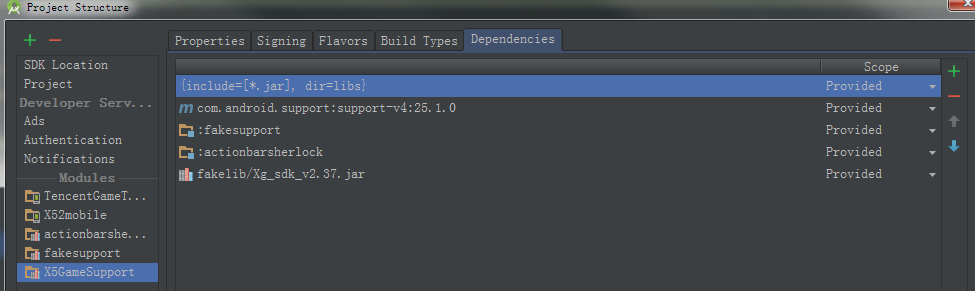
杨煜



1.主要实现核心:

通过使用ClassLoader读取apk包里面的dex包和lib包并且缓存到运行目录进行激活.

使用该ClassLoader才能正常加载apk里的class

1. 拉起步骤:
   1. 将apk放到手机sd卡内(可任意目录.)
   2. 在助手工程任意地方可调用(只要登录了助手)X52LaunchHelper.ins().launchX5Game(...)来拉起.传入data,和apk路径.
   3. Jar包会读取apk路径进行拉起操作.
   4. Jar包找到apk之后首先会读取apk里面的assets/distribute\_path.txt文件,寻找X5GameDistribute的包路径进行读取,如若读取不到,则读取默认路径.配置内容:
      1. supportVersion = 当前jar包的版本号,int型
      2. openLog = 是否强制打开日志
      3. playActivityEnterAnimation = 现在暂时没用,等以后如果有策划对Activity跳转需求,可配置
      4. productName = 这个apk在jar包中的名称保存.用来做数据加载的路径定义等等
      5. 现有的测试方式是:点击某一个条目,就能进入游戏(这是以炫舞1的助手为入口,所以现在开发需要同时拥有炫舞1角色和炫舞2角色.不然无法进入.)
2. 现有的接口规定
   1. //depot/products/Project\_X52\_mobile/X52CoadjutantMoblie/X5GameSupport/版本发布/接口1.1.txt
   2. X52Support\_Global 里面的一些key值
3. 如何打包:
   1. 正常打包TencentGameTools\_release
   2. 如果jar包接口有变动,要修改apk的调用,则需要执行脚本://depot/products/Project\_X52\_mobile/X52CoadjutantMoblie/buildSupport\_and\_move.bat 移动jar包.
   3. 打一个apk包使用脚本//depot/products/Project\_X52\_mobile/X52CoadjutantMoblie/buildX52Apk\_and\_move.bat
   4. debug的工程配置
   5. 使用脚本打包和移动该包其实是调用gradlew的命令执行task,如果命令有问题,可以查找X52mobile的build.gradle, X5GameSupport的build.gradle 中的task
   6. 执行完打包后,即可运行
   7. Release时,如果jar包有变动,则需要给予aar格式的包,该包在X5GameSupport\build\outputs\aar下.
   8. Release时,apk包需要切换到Release并且输入keystone
4. jar包修改除代码意外的规范
   1. 详见//depot/products/Project\_X52\_mobile/X52CoadjutantMoblie/X5GameSupport/规范.txt
5. 工程依赖关系
   1. 
   2. 
   3. 
6. activity的返回和跳转机制.
   1. 现在的activity虽然在助手侧的代理注册不是single的.但是在管理方面是必须要保持single,因为对于安卓而言,每个activity其实都是Px52GameActivity,只是里面的显示内容不同而已.所以在策划进行需求的时候需要将这一点考虑在内.
   2. 由于apk里面的activity都是jar包里面手动new出来,并不是系统生成,所以可能有些生命周期不调用,需要谨慎使用.
7. dump收集
   1. 在appdump的时候,需要收集日志和crash信息.可以在手机的根目录/qqx52\_community/dumpX.zip中获取, 其中X是0-4 一次dump一次zip
8. 一些问题:

如果碰到jar包冲突,将x52mobile中的jar包从compile变成provided

所有activity的Context为ResourceContext

报找不到类,但是明明在,肯定是继承关系出错.看是不是因为打包后,该类消失,或者包路径有问题等等.